

AD-A167 513

THE COMPUTATIONAL COST OF SIMPLEX SPLINE FUNCTIONS(U)
WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER
T A GRANDINE MAR 86 MRC-TSR-2926 DAAG29-80-C-0041

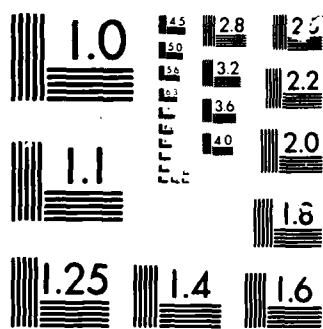
1/1

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY

CHART

AD-A167 513

2

MRC Technical Summary Report #2926

THE COMPUTATIONAL COST OF SIMPLEX
SPLINE FUNCTIONS

Thomas A. Grandine

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

March 1986

(Received March 12, 1986)

DTIC
ELECTE
MAY 21 1986
S D

DTIC FILE COPY

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

National Science Foundation
Washington, DC 20550

86 5 20 127

UNIVERSITY OF WISCONSIN - MADISON
MATHEMATICS RESEARCH CENTER

THE COMPUTATIONAL COST OF SIMPLEX SPLINE FUNCTIONS

Thomas A. Grandine^{1,2}

Technical Summary Report #2926

March 1986

ABSTRACT

This paper presents a negative result concerning the stable evaluation of simplex spline functions. It has been conjectured that a great deal of computational effort might be saved by implementing the recurrence relation for these functions in a clever way. The main result of this paper is that there is no clever way to implement the recurrence relation once the standard recipe for constructing spaces of simplex spline functions has been followed.

AMS(MOS) Subject Classifications: 41A15, 65D07

Key Words: B-spline, simplex spline, spline function, multivariate, recurrence relation

Work Unit Number 3 - Numerical Analysis and Scientific Computing

¹Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

²This material is based upon work supported by the National Science Foundation under Grant No. DMS-8210950, Mod. 4.

SIGNIFICANCE AND EXPLANATION

All computational schemes for simplex spline functions to date rely both on the recurrence relation for these functions and the standard construction of the simplex spline basis. Under these conditions, for numerical methods for computing simplex spline functions to be as useful as possible, it is necessary to find ways of implementing the recurrence relation as efficiently as possible. This paper shows that "as efficiently as possible" is still not very efficient. This implies that truly fast algorithms (which have not yet been developed) will have to either abandon the recurrence relation or the standard construction of the simplex spline basis.

Efficiently as possible - to be analyzed

| | |
|--------------------|--|
| Accession For | |
| NTIS CRA&I | <input checked="checked" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By | |
| Distribution / | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |



The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

THE COMPUTATIONAL COST OF SIMPLEX SPLINE FUNCTIONS

Thomas A. Grandine^{1,2}

Since de Boor defined the multivariate simplex spline in [B76], many of its mathematical properties have been studied and reported ([B82], [D79], [M80], ...). However, numerical methods for evaluating these functions have not been studied. The goal of this paper is to produce an algorithm which evaluates simplex spline functions, i.e. linear combinations of simplex splines. This can be done in a straightforward manner by using the evaluation algorithm for individual simplex splines, described in [G84], to evaluate all non-zero simplex splines at any point x , then summing up the values using the spline function coefficients as weights. However, since this algorithm is based on the recurrence relation for simplex splines, [M80], it seems a wasteful approach. In particular, many of the lower degree simplex splines which need to be evaluated during this process may well be evaluated multiple times. It seems reasonable (and has been conjectured by Dahmen and Micchelli [DM81] and perhaps others) that a great deal of computational effort might be saved by taking advantage of this multiplicity. This paper makes an attempt to do just that.

In order to begin this attempt, the recurrence relation is needed. This is given by

M80

$$M(x|\sigma_0, \sigma_1, \dots, \sigma_n) = \sum_{i=0}^n \alpha_i M(x|\sigma_0, \sigma_1, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_n), \quad (1)$$

¹Sponsored by the United States Army under Contract No. DAAG29-80-C-0041.

²This material is based upon work supported by the National Science Foundation under Grant No. DMS-8210950, Mod. 4.

where

$$x = \sum_{i=0}^n \alpha_i P \sigma_i.$$

Here Pz means the first m components of the vector z .

The second necessary step is to clarify just which simplex splines are the correct ones to deal with when applying this recurrence relation. This requires the construction of an appropriate simplex spline basis. Höllig [H82] and Dahmen and Micchelli [DM82] have simultaneously and independently given versions of one such construction. The Höllig version of the construction is summarized here:

To construct a simplex spline basis over $\Omega \subseteq \mathbf{R}^m$, start with a **triangulation** of Ω .

A triangulation of a set Ω is a collection T of sets τ such that

- (i) $\text{vol}[\tau] > 0$
- (ii) $\cup_{\tau \in T} [\tau] = \Omega$
- (iii) $[\tau] \cap [\tau'] = [\tau \cap \tau'], \quad \tau, \tau' \in T$
- (iv) $\#\tau = m + 1, \quad \tau \in T,$

where $[A]$ is the convex hull of A and $\#A$ is the cardinality of A . Thus, each $[\tau]$ is a simplex in Ω . Consider the **slab** $\Omega \times \Sigma_k := \{(u, v) | u \in \Omega, v \in \Sigma_k\}$, where Σ_k is the standard unit simplex in \mathbf{R}^k . The construction boils down to triangulating this slab and using the vertices of each of the resulting simplices as knot sets for the simplex splines.

The slab can be triangulated as follows: For each $\tau \in T$, consider the following grid:

$$\begin{array}{cccc} t_{0,k} & t_{1,k} & \dots & t_{m,k} \\ \vdots & \vdots & \ddots & \vdots \\ t_{0,1} & t_{1,1} & \dots & t_{m,1} \\ t_{0,0} & t_{1,0} & \dots & t_{m,0} \end{array}$$

Here, t_{ij} is intended to represent (t_i, e_j) , where t_i is a point in τ , and e_j is the j -th standard unit vector (with e_0 being 0). Hence t_{ij} is a vertex of $\tau \times \Sigma_k$. (This is not quite true. In the actual construction, the knots are **pulled apart**. This means that t_{ij} is not really (t_i, e_j) , but rather some point which is close to (t_i, e_j) . The actual details are quite interesting and useful, but not of importance here). Consider all non-decreasing paths from $t_{0,0}$ to $t_{m,k}$. Such a path is denoted by $\sigma_0, \sigma_1, \dots, \sigma_n$, where $\sigma_0 = t_{0,0}$, $\sigma_n = t_{m,k}$, and, if $\sigma_\ell = t_{ij}$, then $\sigma_{\ell+1}$ is either $t_{i+1,j}$ or $t_{i,j+1}$. The set of all paths through the grid for τ for all $\tau \in T$ give rise to a simplex spline basis.

Thus, each of the knots, σ_i , can be classified according to where it was found along the path from $t_{0,0}$ to $t_{m,k}$. Specifically, given $\sigma_\ell = t_{i,j}$, then σ_ℓ is called a **horizontal knot** if $\sigma_{\ell-1} = t_{i-1,j}$ and $\sigma_{\ell+1} = t_{i+1,j}$. If $\sigma_{\ell-1} = t_{i,j-1}$ and $\sigma_{\ell+1} = t_{i,j+1}$, then σ_ℓ is called a **vertical knot**. All other knots are called **corner knots**. Note that these names are specific to a given simplex spline, i.e. a knot which is of one type in one simplex spline may be of a different type in a different simplex spline. This classification of knots has no significance except to relate the simplex splines to the paths from which they were derived. The paths provide the key to piecing lower degree simplex splines together, as will be made clear shortly.

Definition: The simplex spline $M_i := M(\cdot | \sigma_0, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_n)$ is said to **contribute** to the simplex spline $M(\cdot | \sigma_0, \dots, \sigma_n)$ if, in (1), M_i necessarily has a non-zero coefficient corresponding to it.

In order to prove the upcoming theorem in this paper, two more concepts are needed. The first notion is that of a cut corner. A path with a cut corner is obtained from a path $\sigma_0, \dots, \sigma_n$ by removing a corner point. This amounts to finding a path through the grid

consisting of only horizontal and vertical moves, except for one place, arbitrarily chosen, where a diagonal move is allowed. The second notion has to do with the grids themselves. In addition to the grid for τ , a grid may also be considered for $\tau \cap \tau'$. If $\tau \neq \tau'$, then this grid will have fewer than $m + 1$ columns, and a path through the grid will only produce a simplex spline of degree $k + c - m - 1$, where c is the number of columns of the grid.

Theorem 1: Assume that $t_{i,j} \neq t_{i',j'}$ unless $i = i'$ and $j = j'$. Then the only simplex splines of degree $k - 1$ which can contribute to two simplex splines of degree k are those which come from paths with cut corners and those which come from paths corresponding to grids for $\tau \cap \tau'$, where $\# \tau \cap \tau' = m$. There are no simplex splines of degree $k - 1$ which contribute to three or more simplex splines of degree k .

Proof: Take any simplex spline of degree k , say M , deriving from a path corresponding to the grid for τ . Then the only simplex splines which can contribute to M are those obtained by deleting a knot, say σ_j , along that path. Suppose σ_j is a vertical knot. Then $M_j := M(-|\sigma_0, \dots, \sigma_{j-1}, \sigma_{j+1}, \dots, \sigma_n)$ can only contribute to M , since the only way to get from σ_{j-1} to σ_{j+1} along a path is through σ_j . Suppose σ_j is a corner knot. Assume that $\sigma_{j-1} = t_{j-\ell-1, \ell}$. Then, since σ_j is a corner knot, $\sigma_{j+1} = t_{j-\ell, \ell+1}$. Thus, the only way to get from σ_{j-1} to σ_{j+1} along a path is through either $t_{j-\ell-1, \ell+1}$ or $t_{j-\ell, \ell}$, one of which must be σ_j . Thus, M_j can contribute to at most two simplex splines of degree d . Lastly, suppose σ_j is a horizontal knot. Then M_j comes from a path along a grid where the column corresponding to σ_j is deleted. Suppose there exists τ' such that $\tau \cap \tau'$ provides just such a grid. Then M_j contributes both to M and to the corresponding simplex spline from the grid for τ' , but to no others. Otherwise, M_j just contributes to M . This proves the theorem.

A computer code has been written using this idea. The same approach to implementing the recurrence relation that is employed in G84 is used here. Here, however, it is best to have non-zero as few of the α_i corresponding to vertical knots as possible. Unfortunately, including this condition results in a problem whose computational complexity is equivalent to that of the travelling salesman problem. Since the goal here is to write a fast code, the simple heuristic device of steering the method toward corner and horizontal knots whenever possible is employed. This does not behave perfectly, but it does function adequately, and the non-zero α_i obtained usually correspond to corner and horizontal knots when such solutions are possible. Additionally, the code keeps track of which lower degree simplex splines have already been computed so that their results can be reused if possible. A hashing scheme has been implemented to provide nearly direct access to the values of these already computed simplex splines.

Unfortunately, the code behaves quite poorly in practice. In actual comparison with the more naive approach, i.e., the one in which no intermediate results are saved and all simplex splines are computed one at a time, the naive approach is the clear and decisive winner. The clever approach takes approximately three times as much CPU time, and this ratio seems not to change with either the degree of the simplex splines or the complexity of the knots, at least through seventh degree polynomials in the bivariate case!

The explanation for this lies in the excessive overhead involved in keeping track of the lower degree intermediate results. The only way to distinguish between values of different simplex splines is to somehow distinguish between the simplex splines themselves. This can only be done by enumerating the knots which make up the simplex spline. Thus, the amount of overhead increases with the degree of the simplex splines. This means that the

payoff that might be expected by making the degree of the simplex splines sufficiently large does not actually occur.

What's worse, Theorem 1 says, in some sense, that this cannot be improved on. While other programmers might be able to improve the code sufficiently to make it competitive, it cannot be done in such a way that the performance of the naive method can be bettered by more than a factor of 2^{k-1} . Indeed, such a program would have to have some overhead not present in the naive code, and there will usually be lower degree simplex splines which contribute to only one higher degree simplex spline present. Thus, a factor of 2^{k-1} improvement is really far too much to hope for.

References

- [B76] de Boor, Carl. "Splines as linear combinations of B-splines," in *Approximation Theory II*, G. G. Lorentz, C. K. Chui, and L. L. Schumaker, eds., Academic Press, pp. 1-47.
- B82] de Boor, Carl. "Topics in Multivariate Approximation Theory," in *Topics in Numerical Analysis*, P. Turner, ed. *Springer Lecture Notes in Mathematics* **965**, 1982, pp. 39-78.
- D79] Dahmen, W. "Multivariate B-splines - Recurrence Relations and Linear Combinations of Truncated Powers," in *Multivariate Approximation Theory*, W. Schempp and K. Zeller, eds., Basel: Birkhäuser, pp. 64-82.
- [DM81] Dahmen, W., and Micchelli, C. A. "Numerical Algorithms for Least Squares Approximation by Multivariate B-splines," in *Numerical Methods of Approximation Theory*, Vol. 6, ed. by Collatz, Meinardus, and Werner, Basel: Birkhäuser, 1981.
- [DM82] Dahmen, W., and Micchelli, C. A. "On the Linear Independence of Multivariate B-splines I., Triangulation of Simplices," *SIAM Journal of Numerical Analysis* **19**, pp. 993-1012.
- [G84] Grandine, Thomas. "The Stable Evaluation of Multivariate B-splines," Mathematics Research Center TSR #2744, 1984.
- [H82] Höllig, Klaus. "Multivariate Splines," *SIAM Journal of Numerical Analysis* **19**, pp. 1013-1031.

- M80 Micchelli, C. A. "A Constructive Approach to Kergin Interpolation in \mathbf{R}^k : Multivariate B-splines and Lagrange Interpolation." *Rocky Mountains J. Math.* **10**, pp. 485-497.

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|--|
| 1. REPORT NUMBER 2926 | 2. GOVT ACCESSION NO. AD-A121513 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) THE COMPUTATIONAL COST OF SIMPLEX SPLINE FUNCTIONS | | 5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Thomas A. Grandine | | 8. CONTRACT OR GRANT NUMBER(s) DAAG29-80-C-0041 DMS-8210950, Mod. 4 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of 610 Walnut Street Madison, Wisconsin 53705 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 3 - Numerical Analysis and Scientific Computing |
| 11. CONTROLLING OFFICE NAME AND ADDRESS See Item 18 below. | | 12. REPORT DATE March 1986 |
| | | 13. NUMBER OF PAGES 8 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) UNCLASSIFIED |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES U. S. Army Research Office P. O. Box 12211 Research Triangle Park North Carolina 27709 National Science Foundation Washington, DC 20550 | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) B-spline simplex spline spline function multivariate recurrence relation | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper presents a negative result concerning the stable evaluation of simplex spline functions. It has been conjectured that a great deal of computational effort might be saved by implementing recurrence relation for these functions in a clever way. The main result of this paper is that there is no clever way to implement the recurrence relation once the standard recipe for constructing spaces of simplex spline functions has been followed. | | |

END

FILMED

6-86

DTIC